UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/618,367	07/18/2000	Stan Jirman	APLE.P0005	8886

62224 7590 06/09/2009

ADELI & TOLLEN, LLP
11940 San Vicente Blvd., Suite 100
LOS ANGELES, CA 90049

EXAMINER	
CAO, DIEM K	

ART UNIT	PAPER NUMBER
2194	

MAIL DATE	DELIVERY MODE
06/09/2009	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	09/618,367	JIRMAN ET AL.	
	Examiner	Art Unit	
	DIEM K. CAO	2194	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *26 May 2009*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-20 and 23-50* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-20,23-50* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-20 and 23-50 are pending.

Response to Amendment

2. Applicant's request for reconsideration of the finality of the rejection of the last Office action is persuasive and, therefore, the finality of that action is withdrawn.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 39, 42 and 44 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wilner et al. (US 5,872,909) in view of Teegan et al (US 7,243,271 B1).

As to claim 39, Wilner teaches a method of logging events for an application (logs events which occurs in the target software; abstract and col. 18, lines 17-25), the method comprising:

identifying a set of events generated by the application (the user can log application-specific events; col. 18, lines 16-26 and col. 19,lines 3-9),

hierarchical logging the identified set of events, wherein at least one event in the hierarchical comprises a sub-event (Reconstruction of Target State on Host, event stack, levels 1-3; col. 19, line 46 – col. 20, line 25), wherein the identifying and the logging are performed by an

event logging mechanism on a single computer on which the application executes (Event logging; col. 8, line 52, col. 9, lines 9-13 and col. 16, lines 15-28 and a single computer runs both the target and host software; col. 4, lines 51-67 and abstract).

Wilner does not teach the event logging mechanism running independently from the application. Wilner rather teaches the part of event logging mechanism is instrumented the application to performed the identifying and logging steps (event logging is not started when the target is booted, event logging is initiated by the user; col. 9, lines 9-15, when instrumentation is turned on, a series of a particular event type can identify all the tasks running at the time the instrumentation was turned on; col. 12, lines 16-18, and instrumentation routine which logs information about the object when the routine is called; col. 15, line 30 – col. 16, line 66). Teegan teaches the event logging mechanism running independently from the application (an architecture … transparently monitoring interactions with software object … the developer of the monitored software object can write code for the monitored software object without knowledge of the enterprise software management arrangement or architecture; col. 10, lines 7-57).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply the teaching of Teegan to the system of Wilner because Teegan teaches a method that allows monitor software transparently, and overcome all the various problems involve when using intrusive technique (col. 2, line 55 – col. 3, line 18 and lines 28-40).

As to claim 42, Wilner teaches a computer comprising storage for:

- a foundation layer upon which applications are executed (operating system 18; col. 4, lines 36-41); and

- an event logging mechanism for execution on the foundation layer (Event logging; col. 8, line 52, col. 9, lines 9-13 and col. 16, lines 15-28), for functioning interoperably with the application (the event logging is turned on by a user while the application is already run; col. 9, lines 9-13 and col. 16,line 3-28), the mechanism for:

- identifying a set of events for an application executing on the foundation layer (the user can log application-specific events; col. 18, lines 16-26 and col. 19,lines 3-9),

- generating an event log to record the identified event data (Each event as logged has a fixed format … parameter array; col. 11, lines 39-41 and Reconstruction of Target State on Host; col. 19, line 46 – col. 20, line 15), and

- analyzing the event data (a user may examiner an event in more detail … the task ID, etc; col. 8, lines 33-46 and col. 20, lines 16-25 and abstract), wherein the application does not generate an event log (inherent from the event log is generated from the event logging mechanism).

Wilner does not teach the event logging mechanism running separately from the application. Wilner rather teaches the part of event logging mechanism is instrumented the application to performed the identifying and logging steps (event logging is not started when the target is booted, event logging is initiated by the user; col. 9, lines 9-15, when instrumentation is turned on, a series of a particular event type can identify all the tasks running at the time the instrumentation was turned on; col. 12, lines 16-18, and instrumentation routine which logs information about the object when the routine is called; col. 15, line 30 – col. 16, line 66). Teegan teaches the event logging mechanism running independently from the application (an architecture … transparently monitoring interactions with software object … the developer of the

monitored software object can write code for the monitored software object without knowledge of the enterprise software management arrangement or architecture; col. 10, lines 7-57).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply the teaching of Teegan to the system of Wilner because Teegan teaches a method that allows monitor software transparently, and overcome all the various problems involve when using intrusive technique (col. 2, line 55 – col. 3, line 18 and lines 28-40)

As to claim 44, Wilner teaches wherein the event-logging mechanism comprises analyzing the event log according to hierarchical and contextual grouping (col. 19, line 46 – col. 20, line 25).

5. Claims 1-4, 19, 20, 30, 32-36, 40, 41, 43, 46, 47, 49 and 50 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wilner et al. (U.S. 5,872,909) in view of Teegan et al (US 7,243,271 B1) further in view of Ward et al. (U.S. 2002/0083217 A1).

As to claim 1, Wilner teaches a method comprising: for an event to be logged that has not yet been logged within an application (logs events which occurs in the target software; abstract and col. 18, lines 17-25):

- creating an event object (event; col. 18,lines 32-34), the event object occupying a memory space that is independent of said application (memory 20 includes a buffer for storing logs of events 22, host 16 includes a memory 24 with reconstructed data fields 26; col. 4, lines 36-46);

- logging within the event object start time, and information regarding the event (Each event as logged has a fixed format … parameter array; col. 11, lines 39-41);
- analyzing at least one of the start time and information regarding the event (a user may examiner an event in more detail … the task ID, etc; col. 8, lines 33-46);
- wherein the creating, the logging, and the reviewing are performed by an event logging mechanism on a single computer on which the application executes (a single computer runs both the target and host software; col. 4, lines 51-67 and abstract).

Wilner does not explicitly teach logging within the event object end time and reviewing the end time, and the event logging mechanism running independently from the application.

However, Ward teaches logging both start time and end time for an event (time start, time end; page 7, paragraphs 73-74).It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply the teaching of Ward to the system of Wilner because Wilner teaches having both start time and end time enables a user to perform desired evaluation on a currently executing application in real-time without adversely affecting the performance of the computer system (page 2, paragraph 17).

Wilner and Ward do not teach the event logging mechanism running independently from the application. Wilner rather teaches the part of event logging mechanism is instrumented the application to performed the identifying and logging steps (event logging is not started when the target is booted, event logging is initiated by the user; col. 9, lines 9-15, when instrumentation is turned on, a series of a particular event type can identify all the tasks running at the time the instrumentation was turned on; col. 12, lines 16-18, and instrumentation routine which logs information about the object when the routine is called; col. 15, line 30 – col. 16, line 66).

Teegan teaches the event logging mechanism running independently from the application (an architecture … transparently monitoring interactions with software object … the developer of the monitored software object can write code for the monitored software object without knowledge of the enterprise software management arrangement or architecture; col. 10, lines 7-57).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply the teaching of Teegan to the system of Wilner because Teegan teaches a method that allows monitor software transparently, and overcome all the various problems involve when using intrusive technique (col. 2, line 55 – col. 3, line 18 and lines 28-40).

As to claim 2, Wilner teaches checking whether event logging has been turned on for the event. However, Wilner teaches the logging mechanism can be turned on or off by user (start event logging, stop event logging; col. 16, lines 15-28). Ward teaches checking whether event logging has been turned on for the event (pages 7-8, paragraph [0079]). It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply the teaching of Ward to the system of Wilner because Ward teaches a method that logging for only certain events that are interested, not all the events occur in the system (page 6, paragraph [0061]).

As to claim 3, Wilner as modified by Ward teaches wherein the creating and the logging are performed for each event having event logging turned on, wherein a plurality of event objects are created and logged for a plurality of events (see Ward: page 6, paragraph [0061] and page 8, paragraphs [0082]-[0083]).

As to claim 4, Wilner teaches wherein the reviewing comprises analyzing the event object (logic analyzer function on the real-time software; abstract and a user may examiner an event in more detail … the task ID, etc; col. 8, lines 33-46). Wilner does not explicitly teach analyzing the event object after event logging is turned off. However, Wilner teaches the logging mechanism is turned off (stop event logging; col. 16, lines 26-28), and reviewing and analyze the software after all of the interested events have been logged (col. 19, line 32 – col. 20, line 15). It would have been obvious that the analyzing step could have occurred after the logging mechanism is turned off.

As to claim 19, it is the same as the method claim 1 above except this is a computer product claim, and is rejected under the same ground of rejection.

As to claim 20, Wilner teaches the event logging mechanism is configured to analyze the event objects based upon hierarchical and contextual grouping (col. 19, line 46 – col. 20, line 25).

As to claim 30, see rejection of claim 1 above. Wilner further teaches storing the event object in a first memory space that is uniquely allocated for the event logging method, the first memory space separate from a second memory space allocated for the plurality of applications (col. 4, lines 36-44).

As to claims 32-33, see rejections of claims 2 and 4 above.

As to claim 34, Wilner teaches the memory space occupied by the event log is within memory space that has been allocated solely to the event logging mechanism (buffer; col. 4, lines 39-40).

As to claim 35, Wilner teaches wherein the events that are logged by the event logging mechanism have not been previously logged by any other application (col. 18, lines 16-27).

As to claim 36, Wilner teaches the information placed in the event log is first logged by the event logging mechanism (col. 18, lines 28-34).

As to claim 40, see rejections of claims 2-3 above.

As to claim 41, see rejection of claim 4 above.

As to claim 43, Wilner teaches storing, for each event to be logged, a start and information regarding the event (Each event as logged has a fixed format … parameter array; col. 11, lines 39-41).

Wilner does not explicitly teach storing end time. However, Ward teaches logging both start time and end time for an event (time start, time end; page 7, paragraphs 73-74).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply the teaching of Ward to the system of Wilner because Wilner teaches having both start time and end time enables a user to perform desired evaluation on a currently executing application in real-time without adversely affecting the performance of the computer system (page 2, paragraph 17).

As to claim 46, Wilner teaches a method comprising:

identifying a set of events for an application (the user can log application-specific events; col. 18, lines 16-26 and col. 19,lines 3-9);

analyzing the set of events identified for the application (a user may examiner an event in more detail … the task ID, etc; col. 8, lines 33-46 and col. 20, lines 16-25 and abstract and abstract), wherein each event comprises at least a start time (Each event as logged has a fixed format … parameter array; col. 11, lines 39-41);

grouping the set of events based on the analysis of the set of events (task context; col. 19, line 46 – col. 20, line 15);

generating a display the set of events based on the grouping (see Fig. 14 and associated text), wherein the identifying, analyzing, grouping, and generating are performed on a single computer on which the application runs (Event logging; col. 8, line 52, col. 9, lines 9-13 and col. 16, lines 15-28 and a single computer runs both the target and host software; col. 4, lines 51-67 and abstract), the event analysis mechanism running independent from the application on the computer (the event logging is turned on by a user while the application is already run; col. 9, lines 9-13 and col. 16,line 3-28).

Wilner does not explicitly teach end time. However, Ward teaches logging both start time and end time for an event (time start, time end; page 7, paragraphs 73-74).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply the teaching of Ward to the system of Wilner because Wilner teaches having both start time and end time enables a user to perform desired evaluation on a currently executing application in real-time without adversely affecting the performance of the computer system (page 2, paragraph 17).232

As to claim 47, Wilner teaches the set of events are identified by an event logging mechanism prior to performing the analyzing, grouping and generating (col. 18, lines 16-34).

As to claim 49, Wilner teaches wherein the grouping comprises grouping the set of events into hierarchy, wherein at least one event in the hierarchy includes a sub-event (Reconstruction of Target State on Host, event stack, levels 1-3; col. 19, line 46 – col. 20, line 25).

As to claim 50, Wilner as modified by Ward teaches computing statistics using the start time and the end time of the set of events, wherein the generating comprises including the statistics in the display (see Wilner: col. 3, lines 3-4 and col. 20, lines 16-25 and col. 8, lines 19-32) and (see Ward: page 7, paragraph 74 and page 8, paragraph 82).

6. Claims 10, 12-14 and 45 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wilner et al. (U.S. 5,872,909) in view of Teegan et al (US 7,243,271 B1) further in view of Kaler et al. (U.S. 6,742,143 B2).

As to claim 10, Wilner teaches a computer comprising computer readable storage for storing:

- a foundation layer upon which applications are built or executed (operating system 18; col. 4, lines 36-41); and

- an event logging mechanism created by the foundation layer (Event logging; col. 8, line 52, col. 9, lines 9-13 and col. 16, lines 15-28), the logging mechanism executing independently of the application (the event logging is turned on by a user while the application is already run; col. 9, lines 9-13 and col. 16,line 3-28), the mechanism for:

- identifying a set of events for an application executing on the foundation layer (the user can log application-specific events; col. 18, lines 16-26 and col. 19,lines 3-9),

- generating an event log for the application (Reconstruction of Target State on Host; col. 19, line 46 – col. 20, line 15), and

- analyzing the event log (a user may examiner an event in more detail … the task ID, etc; col. 8, lines 33-46 and col. 20, lines 16-25 and abstract), the event log generated without referencing any event logs of the application (inherent from the event log is generated from raw event data; col. 19, lines 32-33 and 46 – col. 20, line 15), wherein the event logging mechanism performs the identifying, generating, and analyzing on the computer on which the application executes (Event logging; col. 8, line 52, col. 9, lines 9-

13 and col. 16, lines 15-28 and a single computer runs both the target and host software; col. 4, lines 51-67 and abstract).

Wilner does not explicitly teach wherein each of the events is designated an enabled/disabled status, wherein a disabled status disables all logging for an event. However, Kaler teaches each of the events is designated an enabled/disabled status (IsActive status … True or False; col. 12, lines 12-15 and lines 46-50), wherein a disabled status disables all logging for an event (IsActive is set to False, and the application never changes; col. 12, lines 42-43 and 53-54). It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply the teaching of Kaler to the system of Wilner because Kaler teaches a method that logging for only certain events that are interested, not all the events occur in the system.

Wilner does not teach the event logging mechanism running independently from the application. Wilner rather teaches the part of event logging mechanism is instrumented the application to performed the identifying and logging steps (event logging is not started when the target is booted, event logging is initiated by the user; col. 9, lines 9-15, when instrumentation is turned on, a series of a particular event type can identify all the tasks running at the time the instrumentation was turned on; col. 12, lines 16-18, and instrumentation routine which logs information about the object when the routine is called; col. 15, line 30 – col. 16, line 66). Teegan teaches the event logging mechanism running independently from the application (an architecture … transparently monitoring interactions with software object … the developer of the monitored software object can write code for the monitored software object without knowledge of the enterprise software management arrangement or architecture; col. 10, lines 7-57).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply the teaching of Teegan to the system of Wilner because Teegan teaches a method that allows monitor software transparently, and overcome all the various problems involve when using intrusive technique (col. 2, line 55 – col. 3, line 18 and lines 28-40).

As to claim 12, Wilner as modified by Kaler teaches the foundational layer is an operating system (see Kaler: operating system; col. 26, line 47).

As to claim 13, Wilner as modified by Kaler teaches the foundational layer is a programmable framework (see Kaler: middleware; col. 26, line 56).

As to claim 14, Wilner teaches the event logging mechanism can be turned on and then off from beyond the execution space of the applications within the foundation layer (col. 16, lines 15-25 and 57-66), the turning on and off separate for each event (The user may also dynamically switch on or off … event logging; col. 16, lines 30-33).

As to claim 45, Wilner does not teach the limitation of claim 45. However Kaler teaches an enable/disable state for each event identified by the application (IsActive status … True or False; col. 12, lines 12-15 and lines 46-50), wherein the disable state precludes any system from creating an event log (If IsActive returns False ... FireEvent; col. 12, lines 42-43 and 53-54), wherein generating an event log is performed for each event having event logging enabled (IsActive status … True or False; col. 12, lines 12-15 and lines 46-50).

7. Claims 15-18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wilner et al. (U.S. 5,872,909) in view of Teegan et al (US 7,243,271 B1) and Kaler et al. (U.S. 6,742,143 B2) further in view of Baker et al. (U.S. 6,611,498 B1).

As to claim 15, Wilner as modified by Kaler does not teach the event logging mechanism can be turned on and turned off and configured using a browser application running on the computer.

However, Kaler teaches the event logging mechanism can be turned on and off and configured using an application (col. 14, lines 63-65). Baker teaches configuring event logging using a web browser application (col. 6, lines 42-67).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to improve the system of Wilner and Kaler by applying the teaching of Baker by implementing the application as a browser application because web browser application has been utilized in the event management and it's a different implementation that developers can choose.

As to claim 16, Wilner teaches the event logging mechanism generates a plurality of event objects (event; col. 18, lines 32-34), and is configured to analyze the event objects (a user may examiner an event in more detail … the task ID, etc; col. 8, lines 33-46) and present the result thereof (col. 20, lines 16-25).

Wilner does not explicitly teach present to the browser application the results thereof. However, Baker teaches present to the browser application the results thereof (col. 8, lines 50-

57).

See rejection of claim 15 above for reason to apply the teaching of Baker to the system of Wilner.

As to claim 17, Wilner as modified by Kaler teaches (see Kaler: col. 16, lines 28-67 and col. 33, lines 23-24, lines 34-50, lines 50-55) the event logging mechanism is configured to analyze the event objects based upon hierarchical (tree, level) and contextual grouping (categories).

As to claim 18, Wilner as modified by Kaler teaches the event logging mechanism is configured to aggregate the event objects deemed identical based upon at least one of hierarchical and contextual grouping (see Kaler: sublevel; col. 33, lines 23-24, lines 34-50, lines 50-55).

8. Claims 5-9, 31, 37 and 38 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wilner et al. (U.S. 5,872,909) in view of Teegan et al (US 7,243,271 B1) and Ward et al. (U.S. 2002/0083217 A1) further in view of Kaler et al. (U.S. 6,742,143 B2).

As to claim 5, Wilner does not teach the limitations of the claim.

However, Kaler teaches wherein analyzing includes allowing user definition of the hierarchical levels of granularity of the events whose event objects are to be analyzed (tree; col. 16, lines 28-39), and allowing user definition of contexts for differentiating repeated occurrences

of events deemed identical by nature of their hierarchical position (event type; col. 16, lines 52-64 and col. 17, line 5-10).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply the teaching of Kaler to the system of Wilner because

As to claim 6, Wilner does not teach the limitations of the claim. However, Kaler teaches wherein analyzing further includes grouping events into their hierarchical subgroups (tree, subtree; col. 16, lines 28-39), and grouping events by their context, if any are defined (event type; col. 16, lines 52-64 and Begin/End, Outbound/Inbound; col. 17, line 5-10). See rejection of claim 5 above for reason to apply the teaching of Kaler to the system of Wilner.

As to claim 7, Wilner does not teach the limitations of the claim. However, Kaler teaches (col. 16, lines 29-39 and col. 25, lines 38-50 and col. 32, lines 10-48) wherein analyzing comprises traversing though the hierarchy of subgroups until the subgroup of finest granularity is traversed (tree, leaves, branches), subdividing the events into further subgroups (branches, categories), computing statistics for each subgroup while traversing (see Fig. 16 and associated text), and displaying the statistics (see Fig. 16 see Fig. 19). See rejection of claim 5 above for reasons to apply the teaching of Kaler to the system of Wilner.

As to claim 8, Wilner does not teach the limitations of the claim. However, Kaler teaches (col. 16, lines 29-39 and col. 25, lines 38-50 and col. 32, lines 10-48) wherein if the subgroup of finest granularity has been traversed, then aggregating events deemed identical by virtue of their

hierarchical position into an aggregate (which events need to be ... grouped and connected; col. 31, lines 30-37), computing statistics for each aggregate (analysis; col. 33, lines 23-40), and displaying the statistics for each the aggregate (display of performance data; col. 33, lines 23-24, lines 34-50, lines 50-55). See rejection of claim 5 above for reasons to apply the teaching of Kaler to the system of Wilner.

As to claim 9, Wilner does not teach the limitations of the claim. However, Kaler teaches wherein the analyzing includes aggregating events deemed identical by virtue of their context into an aggregate (which events need to be ... grouped and connected; col. 31, lines 30-37, sublevels, subdivide; col. 33, lines 50-55), computing statistics for each aggregate (analysis; col. 33, lines 23-40), and displaying the statistics for each aggregate (display of performance data; col. 33, lines 23-24, lines 34-50, lines 50-55). See rejection of claim 5 above for reasons to apply the teaching of Kaler to the system of Wilner.

As to claim 31, Wilner does not teach creating, for the event, an enabled/disabled status wherein the disabled status disables all logging for the event within a system that includes a plurality of applications. However, Kaler teaches creating, for the event, an enabled/disabled status (IsActive status … True or False; col. 12, lines 12-15 and lines 46-50), wherein the disabled status disables all logging for the event within a system that includes a plurality of applications (IsActive is set to False, and the application never changes; col. 12, lines 42-43 and 53-54).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply the teaching of Kaler to the system of Wilner because Kaler teaches a method that logging for only certain events that are interested, not all the events occur in the system.

As to claim 37, see rejection of claim 31 above.

As to claim 38, Wilner as modified by Kaler teaches the creating is done by a foundation layer that is a development framework (see Kaler: middleware; col. 26, line 56).

9. Claim 11 is rejected under 35 U.S.C. 103(a) as being unpatentable over Wilner et al. (U.S. 5,872,909) in view of Teegan et al (US 7,243,271 B1) and Kaler et al. (U.S. 6,742,143 B2) further in view of Ward et al. (U.S. 2002/0083217 A1).

As to claim 11, Wilner teaches the event logging mechanism logs start time and other event information into an event object for each event to be logged (Each event as logged has a fixed format … parameter array; col. 11, lines 39-41).

Wilner does not explicitly teach logging within the event object end time. However, Ward teaches logging both start time and end time for an event (time start, time end; page 7, paragraphs 73-74).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply the teaching of Ward to the system of Wilner because Wilner teaches having both start time and end time enables a user to perform desired evaluation on a currently

executing application in real-time without adversely affecting the performance of the computer system (page 2, paragraph 17).

10. Claims 23-27 and 29 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wilner et al. (U.S. 5,872,909) in view of Teegan et al (US 7,243,271 B1) further in view of Baker et al. (U.S. 6,611,498 B1).

As to claim 23, Wilner teaches a computer comprising computer readable storage for storing:

- a foundation layer upon which applications are built or executed (operating system 18; col. 4, lines 36-41); and

- an event logging mechanism for execution on the foundation layer (Event logging; col. 8, line 52, col. 9, lines 9-13 and col. 16, lines 15-28), the logging mechanism executing independently of the applications (the event logging is turned on by a user while the application is already run; col. 9, lines 9-13 and col. 16,line 3-28), the mechanism for:

- identifying a set of events for an application executing on the foundation layer (the user can log application-specific events; col. 18, lines 16-26 and col. 19,lines 3-9),

- generating a hierarchical event log for the application for display, the event log generated without referencing any event logs of the application and comprising an event in the hierarchy that includes a sub-event(Reconstruction of Target State on Host, event stack, levels 1-3; col. 19, line 46 – col. 20, line 25), wherein the event-logging mechanism performs the identifying and generating on the computer on which the

application executes (Event logging; col. 8, line 52, col. 9, lines 9-13 and col. 16, lines 15-28 and a single computer runs both the target and host software; col. 4, lines 51-67 and abstract).

Wilner does not explicitly teach display in a web browser. However, Baker teaches display in a web browser (col. 6, lines 42-67 and col. 8, lines 50-57). It would have been obvious to one of ordinary skill in the art at the time the invention was made to improve the system of Wilner by applying the teaching of Baker by implementing the display as a browser application because web browser application has been utilized in the event management and it's a different implementation that developers can choose.

Wilner does not teach the event logging mechanism running independently from the application. Wilner rather teaches the part of event logging mechanism is instrumented the application to performed the identifying and logging steps (event logging is not started when the target is booted, event logging is initiated by the user; col. 9, lines 9-15, when instrumentation is turned on, a series of a particular event type can identify all the tasks running at the time the instrumentation was turned on; col. 12, lines 16-18, and instrumentation routine which logs information about the object when the routine is called; col. 15, line 30 – col. 16, line 66). Teegan teaches the event logging mechanism running independently from the application (an architecture … transparently monitoring interactions with software object … the developer of the monitored software object can write code for the monitored software object without knowledge of the enterprise software management arrangement or architecture; col. 10, lines 7-57).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply the teaching of Teegan to the system of Wilner because Teegan teaches a

method that allows monitor software transparently, and overcome all the various problems involve when using intrusive technique (col. 2, line 55 – col. 3, line 18 and lines 28-40).

As to claim 24, Wilner teaches wherein the generating an event log comprises storing, for each event to be logged, a temporal attribute of an event in an event object associated with the event (Each event as logged has a fixed format … parameter array; col. 11, lines 39-41).

As to claim 25, Wilner teaches wherein the event-logging mechanism is further for analyzing the event log according to hierarchical and contextual grouping (col. 19, line 46 – col. 20, line 25).

As to claim 26, Wilner teaches a first area of memory allocated to the first application, a second area of memory allocated to the event logging mechanism, wherein the first area of memory allocated to the application is separate from the second area of memory allocated to the event logging mechanism (col. 4, liens 36-46).

As to claim 27, Wilner does not teach the event logging mechanism can be turned on and turned off and configured using the web browser. Baker teaches configuring event logging using a web browser application (col. 6, lines 42-67). See rejection of claim 23 above for reason to apply the teaching of Baker to the system of Wilner.

As to claim 29, Wilner teaches wherein the foundation layer is an operating system upon which applications are executed (operating system 18; col. 4, lines 36-39).

11. Claim 28 is rejected under 35 U.S.C. 103(a) as being unpatentable over Wilner et al. (U.S. 5,872,909) in view of Teegan et al (US 7,243,271 B1) and Baker et al. (U.S. 6,611,498 B1) further in view of Kaler et al. (U.S. 6,742,143 B2).

As to claim 28, Wilner does not teach the limitations. However, Kaler teaches allowing a user to enable or disable event logging for each event in the set of events (IsActive status … True or False; col. 12, lines 12-15 and lines 46-50), wherein the generating the event log is performed for each event having event logging enabled (col. 12, lines 34-45). It would have been obvious to one of ordinary skill in the art at the time the invention was made to apply the teaching of Kaler to the system of Wilner because Kaler teaches a method that logging for only certain events that are interested, not all the events occur in the system.

12. Claim 48 is rejected under 35 U.S.C. 103(a) as being unpatentable over Wilner et al. (U.S. 5,872,909) in view of Teegan et al (US 7,243,271 B1) and Ward et al. (U.S. 2002/0083217 A1) further in view of Baker et al. (U.S. 6,611,498 B1).

As to claim 48, Wilner does not explicitly teach displaying the groupings of the set of events in a web browser.

However, Baker teaches display event information in a web browser (col. 6, lines 42-67 and col. 8, lines 50-57).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to improve the system of Wilner by applying the teaching of Baker by implementing the display as a browser application because web browser application has been utilized in the event management and it's a different implementation that developers can choose.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to DIEM K. CAO whose telephone number is (571)272-3760. The examiner can normally be reached on Monday - Friday, 7:30AM - 4:30PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Hyung Sough can be reached on (571) 272-6799. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/DIEM K CAO/
Primary Examiner
Art Unit 2194

DC
June 7, 2009